INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Word Sense Disambiguation

L Sai Chaitanya Reddy (2023202023)

Shiv Kumar Modi (2023202024)

Introduction

- WSD is a critical task in Natural Language Processing (NLP) aimed at identifying which sense of a word is used in a sentence when the word has multiple meanings.

- Essential for various NLP applications such as machine translation, information retrieval, and text summarization.

- **Challenges** arise due to polysemy, where words have multiple meanings depending on context.

- **Scope of the Project:** Explore and evaluate machine learning models for WSD, and develop a model from scratch.

- **Dataset Selection:** Utilize SemCor and SemEval datasets known for contextual diversity.

- **Systematic Evaluation:** Assess model performance using standard metrics like accuracy for comparative analysis.

Dataset Overview

- Total Sentences - 37,170 sentences and 2,26,040 sense annotations.
- Semantically Annotated: Sentences are labeled with their intended meanings.
- Diverse Categories: Covers various genres like news, fiction, etc., ensuring broad applicability.
- Data Structure: Two files—semcor.data for sentence and sense ID information, semcor.gold.key for WordNet sense IDs.
- Foundation for WSD: Crucial resource for training and evaluating Word Sense Disambiguation (WSD) models.
- Enables Research: Facilitates exploration of contextual disambiguation methods and model development in NLP.

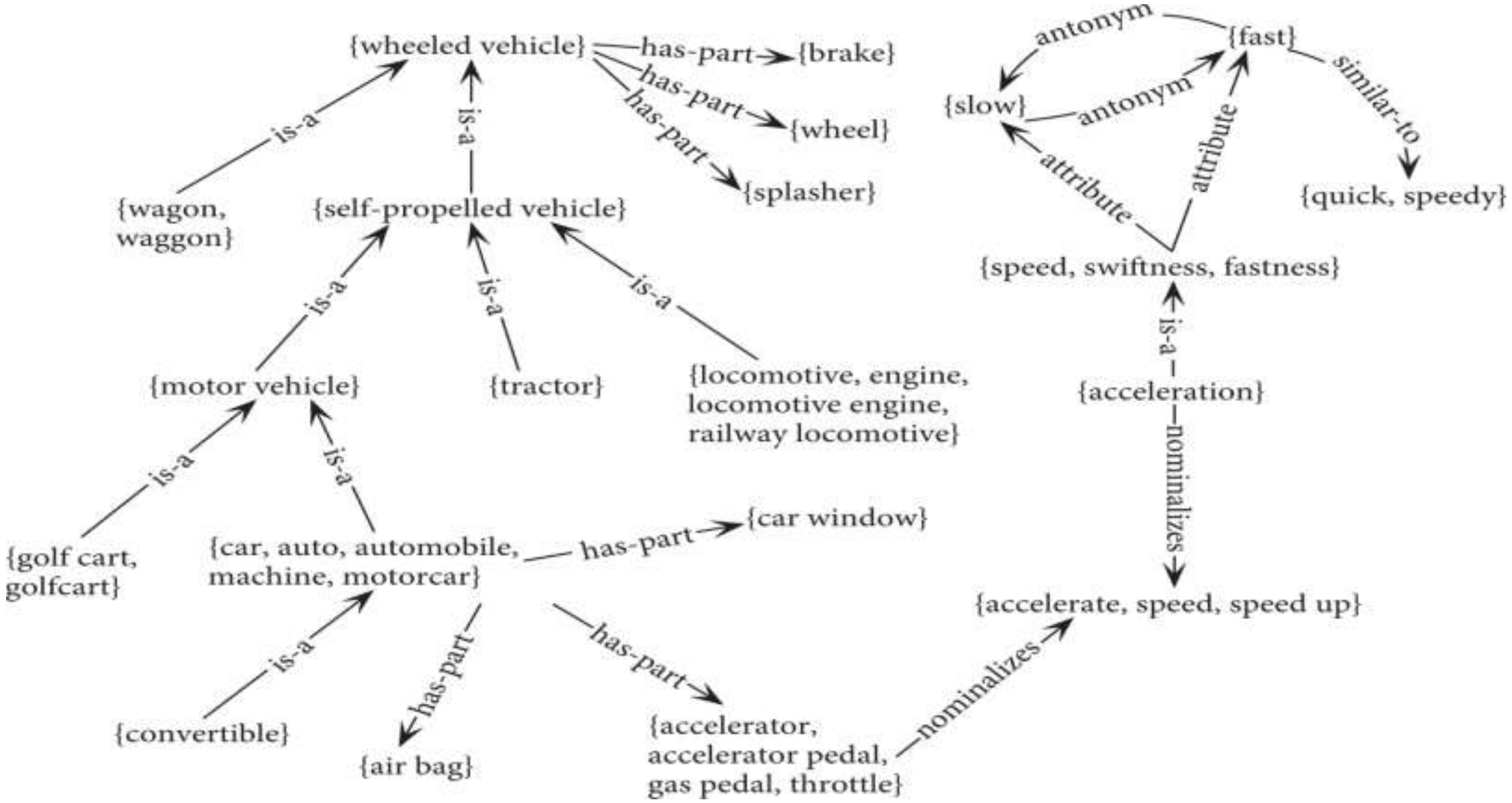
Knowledge Base - WordNet

- Overview: Lexical database of English, organizing words into synsets (sets of synonyms).
- Structure: Arranged as a graph, with synsets as nodes and lexical-semantic relations as edges.
- Synsets: Groups of contextual synonyms representing different senses of a word.
- Lexical-Semantic Relations: Include hypernymy (is-a) and meronymy (part of), forming a hierarchical meaning structure

Key Points:

- Sense Inventory: De facto resource for WSD in English.
- Hierarchical Structure: Facilitates understanding of word meanings and semantic relationships.
- Practical Application: Essential for both defining sense labels and exploring semantic connections in NLP projects.

WordNet viewed as a graph.



1. Approaches - 1 Nearest Sense

Training Phase

Each sentence in the SemCor labeled dataset is passed through a contextual embedding model (BERT-Base-uncased).

Pooling: Summing vector representations from the last four BERT hidden layers.

For each sense s of any word in the corpus, average the contextual representations vi of each token representing that sense to produce a contextual sense embedding V(s)

$$\mathbf{v}_s = \frac{1}{n}\sum_i \mathbf{v}_i \qquad \forall \mathbf{v}_i \in \text{tokens}(s)$$

1. Approaches - 1 Nearest Sense

Testing Phase:

Given a token of a target word t in context:

Compute its contextual embedding t.

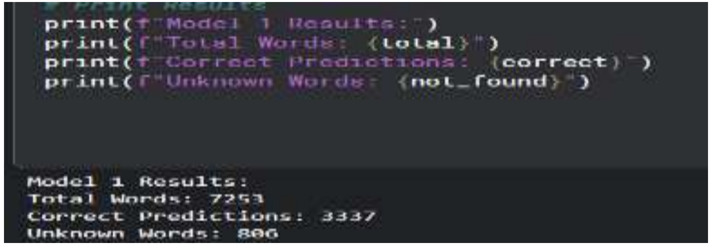
Choose its nearest neighbor sense from the training set based on cosine similarity, selecting the sense whose embedding has the highest cosine similarity with t.

$$\text{sense}(t) = \underset{s \in \text{senses}(t)}{\text{argmax}} \ \text{cosine}(\mathbf{t}, \mathbf{v}_s)$$

- **Approaches - 1 Nearest Sense**

Parameters Used - Embedding Dimension: 300

Evaluation Datasets	Accuracies	Precsion	Recall	F1
Semeval2007	44.09	44	42	43
Semeval2013	52.87	52.9	44.8	48.5
Semeval2015	51.78	59.2	51.9	55.3
Senseval2	53.16	54.8	47.8	51.1
Senseval3	51.18	51.5	48.3	49.9
Concatenated Dataset	51.76	53.4	47.5	50.2



```
  # Print Results
  print(f"Model 1 Results:")
  print(f"Total Words: {total}")
  print(f"Correct Predictions: {correct}")
  print(f"Unknown Words: {not_found}")
```

```
Model 1 Results:
Total Words: 7253
Correct Predictions: 3337
Unknown Words: 806
```

Approaches – 2 Context2Vec

Architecture Overview:

- Bidirectional LSTM (BiLSTM) networks process sentence words from left to right and right to left separately for each word in the sentence.
- LSTM outputs from both directions are concatenated to capture comprehensive sentential context.
- Multi-layer perceptron (MLP) captures complex dependencies between the two sides of the context.
- Joint sentential context around the target word and the target word itself are embedded into the same low-dimensional space.

Context2Vec - Training Objective

Bidirectional LSTM (BiLSTM) Context Representation:

- Utilizes lLS (left-to-right) and rLS (right-to-left) LSTM networks to capture sentence-level context. Shallow bidirectional LSTM context representation for target word wi then concatenates distinct left-to-right/right-to-left word embeddings of sentence words, excluding the target word itself.

$$\mathrm{biLS}(w_{1:n}, i) = \mathrm{lLS}(l_{1:i-1}) \oplus \mathrm{rLS}(r_{n:i+1})$$

Non-linear Transformation:

- Applies a non-linear function on the concatenation of left and right context representations, utilizes Multi Layer Perceptron (MLP) with Rectified Linear Unit (ReLU) activation function for transformation.

$$\mathrm{MLP}(x) = L_2(\mathrm{ReLU}(L_1(x)))$$

Context2Vec - Training Objective

Context2vec Representation

- Defines context2vec's representation of the sentential context c, Represents the entire joint sentential context around the target word.

$$\vec{c} = \text{MLP}(\text{biLS}(w_{1:n}, i)).$$

Learning Objective

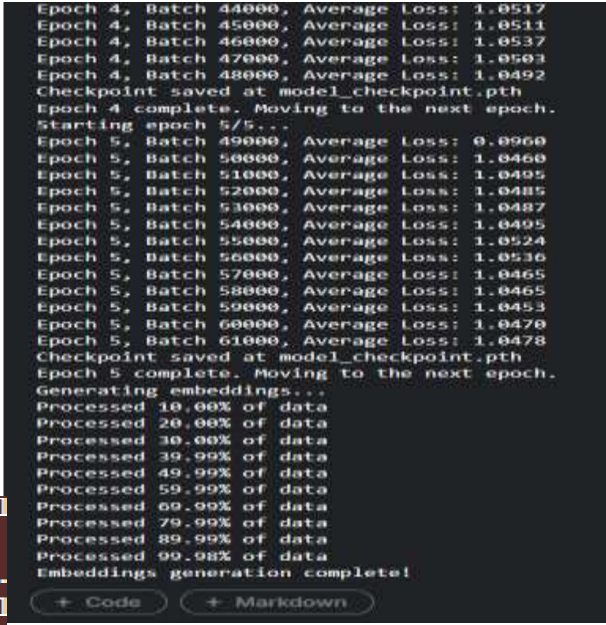
- Learn target word and context rep using word2vec negative sampling objective function.
- Minimizes the difference between the dot product of the target word and context representations and the sigmoid function value.
- Utilizes negative sampling for efficient training, with negative samples sampled from the training corpus.

$$S = \sum_{t,c} \left(\log \sigma(\vec{t} \cdot \vec{c}) + \sum_{i=1}^{k} \log \sigma(-\vec{t_i} \cdot \vec{c}) \right)$$

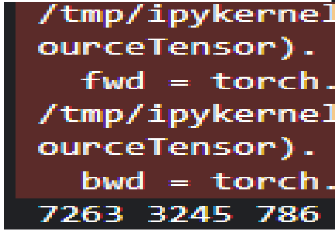
Context2Vec - Results

Parameters Used No. of negative Samples : 5 Embedding No. of Epochs : 3 target_embedding_dim=600, context_embedding_dim=300,lstm_hidden_dim=600, mlp_hidden_dim=1200, batch_size=256

Evaluation Datasets	Accuracies	Precsion	Recall	F1
Semeval2007	48.75	49	47.5	48.2
Semeval2013	52.89	51.1	45.1	47.9
Semeval2015	45.85	53	48.1	50.4
Senseval2	52.98	54.6	48.8	51.5
Senseval3	51.82	51.3	48.8	50
Concatenated Dataset	50.67	52.3	47.8	50



```
Epoch 4, Batch 44000, Average Loss: 1.0517
Epoch 4, Batch 45000, Average Loss: 1.0511
Epoch 4, Batch 46000, Average Loss: 1.0537
Epoch 4, Batch 47000, Average Loss: 1.0503
Epoch 4, Batch 48000, Average Loss: 1.0492
Checkpoint saved at model_checkpoint.pth
Epoch 4 complete. Moving to the next epoch.
Starting epoch 5/5...
Epoch 5, Batch 49000, Average Loss: 0.0960
Epoch 5, Batch 50000, Average Loss: 1.0460
Epoch 5, Batch 51000, Average Loss: 1.0495
Epoch 5, Batch 52000, Average Loss: 1.0485
Epoch 5, Batch 53000, Average Loss: 1.0487
Epoch 5, Batch 54000, Average Loss: 1.0495
Epoch 5, Batch 55000, Average Loss: 1.0524
Epoch 5, Batch 56000, Average Loss: 1.0536
Epoch 5, Batch 57000, Average Loss: 1.0465
Epoch 5, Batch 58000, Average Loss: 1.0465
Epoch 5, Batch 59000, Average Loss: 1.0453
Epoch 5, Batch 60000, Average Loss: 1.0470
Epoch 5, Batch 61000, Average Loss: 1.0478
Checkpoint saved at model_checkpoint.pth
Epoch 5 complete. Moving to the next epoch.
Generating embeddings...
Processed 10.00% of data
Processed 20.00% of data
Processed 30.00% of data
Processed 39.99% of data
Processed 49.99% of data
Processed 59.99% of data
Processed 69.99% of data
Processed 79.99% of data
Processed 89.99% of data
Processed 99.98% of data
Embeddings generation complete!
```



```
/tmp/ipykernel
ourceTensor).
  fwd = torch.
/tmp/ipykernel
ourceTensor).
  bwd = torch.
7263 3245 786
```

Total , Correct , Not_found words

Approaches – 3 GlossBERT

- **GlossBERT** extends BERT by not only considering the contextual sentence but also integrating the definitions (glosses) of words directly into the model, enhancing the model's ability to distinguish the correct sense based on both usage and definition.

- **Benefit** BERT can explicitly model the relationship of a pair of texts, as it is trained on NSP (Next sentence prediction) which has shown to be beneficial to many pairwise natural language understanding tasks. In GlossBERT we construct context-gloss pairs from possible senses of the target word in WordNet, thus treating WSD task as a sentence pair classification problem.

- **Negative sampling:** Negative Sampling is a training strategy used to teach models to distinguish not just the correct answer but also to identify what are incorrect answers.

- **Application to WSD**:In the context of GlossBERT, negative sampling involves presenting both correct sense and multiple incorrect senses during training, enhancing the model's ability to accurately perform sense disambiguation.

Input preparation

- **Context-Gloss Pairs:** The sentence containing target words is denoted as context sentence. For each target word, we extract glosses of all N possible senses of the target word in WordNet to obtain the gloss sentence. [CLS] and [SEP] marks are added to the context-gloss pairs to make it suitable for the input of BERT model.

- **Context-Gloss Pairs with Weak Supervision**: Based on the previous construction method, we add weak supervised signals to the context-gloss pairs.The signal in the gloss sentence aims to point out the target word. Signal is encoded between <target> target word </target>in gloss sentence.

Input preparation

- Pretrained BERT model is fine-tuned with an extra fully connected layer for this binary classification setup.

- This method takes in the final latent embeddings corresponding to [CLS] token and makes a prediction. [CLS] token embeddings gives the representation of the whole *context-gloss* pair along with weak supervision.

- The [CLS] token embeddings in the final layer is used for prediction.

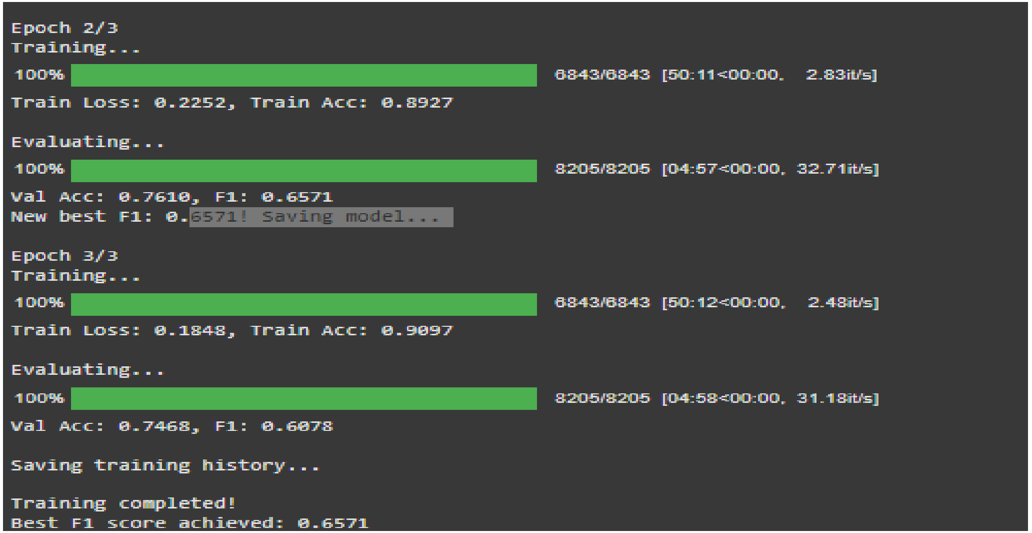
- Binary cross entropy loss function is used.

- $\text{BCE}=-(y\log(p)+(1-y)\log(1-p))$

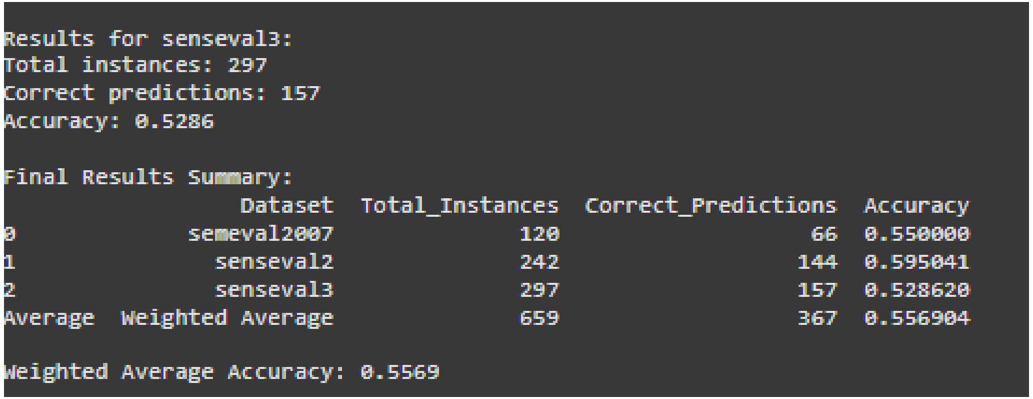
GlossBert - Results

Glossbert(**Sent-CLS-WS**) results

Evaluation Datasets	**Accuracies**
Validation	76%
SenseEval2	60%
SemEval2007	55%
Weighed average	56%



```
Epoch 2/3
Training...
 100%                                         6843/6843 [50:11<00:00,  2.83it/s]
Train Loss: 0.2252, Train Acc: 0.8927

Evaluating...
 100%                                         8205/8205 [04:57<00:00, 32.71it/s]
Val Acc: 0.7610, F1: 0.6571
New best F1: 0.6571! Saving model...

Epoch 3/3
Training...
 100%                                         6843/6843 [50:12<00:00,  2.48it/s]
Train Loss: 0.1848, Train Acc: 0.9097

Evaluating...
 100%                                         8205/8205 [04:58<00:00, 31.18it/s]
Val Acc: 0.7468, F1: 0.6078

Saving training history...

Training completed!
Best F1 score achieved: 0.6571
```



```
Results for senseval3:
Total instances: 297
Correct predictions: 157
Accuracy: 0.5286

Final Results Summary:
                  Dataset  Total_Instances  Correct_Predictions  Accuracy
0             semeval2007              120                   66  0.550000
1               senseval2              242                  144  0.595041
2               senseval3              297                  157  0.528620
Average  Weighted Average              659                  367  0.556904

Weighted Average Accuracy: 0.5569
```

Future Work.

LLMs for WSD

Usage in Zero-Shot or Few-Shot Learning

- **Zero-Shot Learning:** LLMs can be directly queried with prompts that include the word in question along with possible meanings (glosses) to see which context fits best, without any additional training specific to WSD.

- **Few-Shot Learning**: Similar to zero-shot, but providing a few examples of the task completed correctly before asking the model to disambiguate new instances. This can fine-tune the model's predictions based on a small number of examples.

Future Work.

LLMs for WSD

Fine-Tuning on WSD Datasets

•Fine-tuning process involves modifying a pre-trained models by continuing the training process so the model can adjust to the specifics of WSD.

ConSeC:

A novel approach to WSD leveraging a recent re-framing of this task as a text extraction problem, with feedback loop strategy that allows the disambiguation of a target word to be conditioned not only on its context but also on the explicit senses assigned to nearby words.

References

GlossBert : BERT for Word Sense Disambiguation with Gloss Knowledge
https://arxiv.org/pdf/1908.07245

Context2vec : Learning Generic Context Embedding with Bidirectional LSTM

https://aclanthology.org/K16-1006/

Recent Trends in Word Sense Disambiguation: A Survey:
https://www.ijcai.org/proceedings/2021/0593.pdf

Speech and Language Processing textbook by Daniel Jurafsky
https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf

:

Thank you for your Attention.